نسخة أرشيفية معاد بناؤها من النص المفهرس - RECONSTRUCTED_FROM_INDEXED_TEXT

<PARSED TEXT FOR PAGE: 1 / 49>
MOUSSAIF AGI OS / MOUSSAIXT — Reliability-First AGI
1 | ورقة بحثية هندسية 2026 — Moussaif Zouhair / زهير مساعف
الموثوقية كحاكمية تشغيلية في الذكاء الوكيلي العام
Reliability-First AGI (RFAGI) و RAGI2
Reliability Fabric
داخل MOUSSAIF AGI OS / MOUSSAIXT
من صحة الإجابة إلى قابلية إثبات الأثر، والتراجع، والتعافي، والتعلم من الخبرة المتحققة
الحالة المعرفية للورقة
هذه الورقة تقترح إطارًا بحثيًا وهندسيًا جديدًا داخل سلالة MOUSSAIF، ولا تّدعي أن AGI قد تحقق أو أن الموثوقية قدُ حّلت نهائيًا. بعض الأسس مستخرجة مباشرة من corpus المشروع: 4، Hierarchy Evidence، T–0T Authorization، Verification & World-Effect Loop، Observability، Failure Recovery & Rollback، Self-Reconstruction Memory، ونظرية الموثوقية التصعيدية. أما تسميتا RFAGI وRAGI2 وطبقة Fabric Reliability وصيغ القياس الجديدة فهي تركيب مقترح في هذه الورقة ويجب أن يخضع للتنفيذ، المقارنة، ablation، والتكرار.
Moussaif Zouhair / إنجاز وتركيب: زهير مساعف
Arabic RTL / Amiri / A4 — 2026 سبتمبر
مفتاحية كلمات: Rollback، Reversibility، World-Effect Verification، Reliability Governance، Agent Reliability، Verified Learning، HyperSkills، MOUSSAIXT، HAKIM-MCD، MCD/QMCD، Tool Reliability، Observability.
<PARSED TEXT FOR PAGE: 2 / 49>
MOUSSAIF AGI OS / MOUSSAIXT — Reliability-First AGI
2 | ورقة بحثية هندسية 2026 — Moussaif Zouhair / زهير مساعف
الملخص
تتناول هذه الورقة فجوة مركزية في هندسة الأنظمة الوكيلة: ارتفاع القدرة لا يضمن ارتفاع الموثوقية. قد ينجح الوكيل في مهمة معيارية ثم يفشل عند إعادة المحاولة، أو يتخذ مسارًا صحيحًا ظاهريًا لكن دون إثبات الأثر الخارجي، أو ينّفذ خطوة تغّير العالم قبل أن يملك آلية رجوع، أو يدّون تجربة فاشلة في الذاكرة كأنها خبرة صحيحة. تقترح الورقة نقل Reliability من كونها خاصية لاحقة أو مجموعة اختبارات إلى كونها حاكمية تشغيلية ممتدة عبر دورة الإدراك والتخطيط والفعل والتحقق والتعافي والذاكرة. هذا الانتقال يسمى Reliability-First AGI (RFAGI)، بينما يسمى النسيج التنفيذي المقترح Fabric Reliability 2RAGI.
ينطلق البناء من أصول موثقة في MOUSSAIF corpus:« الحالة المرصودة أعلى من الذاكرة أو التوليد»، و«Capability does not imply authorization»، وتعريف World-Effect Loop كسلسلة Proposal → Attempted → Verified → Action → Result → External Effect → Measured، واشتراط Postconditions قابلة للرصد، وسجل Trace يربط mission بالخطة والأداة والدليل وdelta state، ونمذجة الفشل كحالة صريحة مع rollback إلزامي التحقق. كما يستفيد من نظرية الموثوقية التصعيدية التي تصف الموثوقية بأنها صعود من الاتساق إلى الدليل ثم الاعتراض فالاختبار فالأثر العالمي.]1M–5M]
تقارن الورقة هذا البناء باتجاهات حديثة في تقييم الوكلاء. تشير أدبيات 2026 إلى أن تحسن accuracy عبر أجيال النماذج لم يترجم تلقائيًا إلى تحسن مماثل في consistency وrobustness وpredictability، وأن متوسط النجاح يخفي فروقًا مهمة في شدة الخطأ وتكراره وقابلية التنبؤ به. كما تبين τ-bench أهمية التكرار عبر k^pass، ويشدد OSWorld وToolSandbox على based-execution evaluation والحالة الفعلية، بينما تبرز SABER خطورة الأخطاء في actions mutating، وتقدم أعمال rollback الحديثة سنًدا لفكرة أن الاستقلالية يجب أنُ تربط بقابلية التراجع.]1R–12R]
الإسهام المقترح هو:)1(تعريف الموثوقية كـ infrastructure governing لا score وحيد؛)2(ملف موثوقية متعدد المحاور يفصل المعرفة عن التنفيذ والأثر والتعافي والذاكرة؛)3(Fabric Reliability بين التنفيذ والتحقق؛)4(machine state لا تسمح بترقية الفعل إلى نجاح متحقق قبل قراءة البيئة authoritative؛)5(تصعيد مستوى التحقق وفق risk/MCD؛)6(بروتوكول تجريبي من اثني عشر اختبارًا مع ablations؛)7(قاعدة »لا تعلم من trajectory غير متحققة«؛ و)8(تحويل الفشل من استثناء إلى أصل معرفي قابل للتشخيص والتصحيح.
الجملة المؤسسة
النظام الموثوق ليس النظام الذي ينجح كثيرًا فقط؛ بل النظام الذي يعّرف ما يعنيه النجاح قبل الفعل، يثبت ما حدث بعده، يعرف متى

لم ينجح، يحدّ أثر الفشل، يستطيع الرجوع أو التعويض، ثم يتعلم فقط من خبرة تحمل دليلها.
<PARSED TEXT FOR PAGE: 3 / 49>

خريطة الورقة والإسهامات
المخرج السؤال الحاكم المحور
تعريف فجوة Reliability Gap لماذا لا تكفي accuracy/capability؟ المشكلة
RFAGI + خمس نظريات فرعية ما الموثوقية في نظام وكيل؟ النظرية
Fabric Reliability 9.5L أين تسكن الموثوقية؟ المعمارية
Contract Postcondition + Contract Claim كيف نمنع النجاح الوهمي؟ العقود
الفعل ؟mutating actions نضبط كيف Mutation Gate + Reversibility Classifier
التحقق ؟world effect نثبت كيف Independent Verifier + State Diff
الفشل الأعمى؟ retry نمنع كيف Failure Object + Recovery Policy
Memory Experience Verified ماذا يحق للنظام أن يتعلم؟ الذاكرة
القياس
كيف نقيس reliability دون إخفاء tail
؟risks
Reliability Profile + hard gates
12 Falsification/Ablation + Experiments كيف يمكن هدم الإطار؟ التجربة
ترتيب الورقة مقصود: يبدأ من ضبط الادعاء، ثم يعرّف المشكلة، ثم يربطها بأصلها في corpus، ثم يبني النظرية والمعمارية، ثم ينتهي بمقاييس وبرنامج اختبار قابل للفشل. أي عنصر جديد لا يغيّر قرارًا أو اختبارًا أو مسارًا تشغيليًا يجب خفضه من »مكوّن« إلى »تسمية« حتى يثبت فرقًا.
<PARSED TEXT FOR PAGE: 4 / 49>

1. المشكلة البحثية: فجوة القدرة والموثوقية
نجاح الوكيل في benchmark واحد لا يجيب عن أسئلة تشغيلية أكثر صعوبة: هل سيعيد النجاح عند تكرار نفس المهمة؟ هل يحتمل إعادة صياغة الطلب؟ هل يتصرف بصورة سليمة عند timeout أو response ناقص من أداة؟ هل يعرف أنه غير واثق قبل تنفيذ فعل لا رجعة فيه؟ هل يفرق بين »الأداة قالت OK »وبين »الحالة الخارجية أصبحت صحيحة«؟ وهل فشل بسيط في منتصف سلسلة طويلة سيقوده إلى إصلاح موجّه أم إلى تكرار أعمى يزيد الضرر؟ هذه الأسئلة تجعل Reliability أقرب إلى علم لسلوك النظام عبر الزمن والاضطراب والأثر، لا إلى property للنموذج وحده.
العمل الحديث Towards a Science of AI Agent Reliability يبلور المشكلة بدقة: متوسط النجاح يخفي الاتساق عبر runs، والصلابة تحت perturbations، وقدرة الثقة على التنبؤ بالنجاح أو الفشل، وشدة الضرر عندما تقع المخالفة. وقد وجد المؤلفون أن مكاسب reliability عبر 24 شهرًا كانت أبطأ بكثير من مكاسب capability، ما يعني أن scaling أو reasoning الأقوى لا يحلان المسألة آليًا.
MOUSSAIF: observed state + explicit authorization + evidence في invariant مع مباشرة تتقاطع النتيجة هذه[R1]
[M1] output volume. أو model confidence على تتقدم + verification
نعّرف فجوة الموثوقية هنا بوصفها المسافة بين »قدرة النظام على اقتراح أو محاولة عمل« و»قدرته على إنجاز ذلك العمل بصورة قابلة للتكرار، متحققة، محدودة الضرر، قابلة للتفسير والتعافي«. يمكن أن يكون agent قويًا في planning وضعيفًا في checking postcondition؛ قويًا في selection tool وضعيفًا في authorization؛ قويًا في coding وضعيفًا في rollback؛ أو قويًا في الإجابة وضعيفًا في calibration. لذلك لا يصح اختزال الموثوقية في نقطة واحدة.
سؤال البحث المركزي
كيف نبني نظامًا وكيليًا متقدمًا تكون الموثوقية فيه شرطًا حاكمًا للفعل والتعلم، لا اختبارًا يطبق بعد أن تكتمل القدرة؟
<PARSED TEXT FOR PAGE: 5 / 49>

2. المنهج ومراتب الادعاء
تعتمد الورقة منهجا تركيبيًا هندسيًا. القسم الأول derived-Source: يعيد ترتيب عناصر موجودة في MOUSSAIF corpus دون

الادعاء أنها جديدة. القسم الثاني synthesis Proposed: يبني RFAGI و2RAGI كمقترح جديد داخل المشروع. القسم الثالث comparison External: يقارن هذا المقترح بأعمال منشورة. القسم الرابع program Experimental: يحول المقترح إلى فرضيات واختبارات. هذه الحدود مهمة لأن قوة الورقة لا تأتي من قوة الاسم بل من وضوح status لكل دعوى.

المثال في هذه الورقة المعنى الرتبة

نتيجة benchmark لم تنفذ بعد غير متحقق / لا دليل كاف 0E

هدف أو اسم ظهر في سياق المشروع مذكور أو مطلوب 1E

E2 artifact في موجود ملفات في T0–T4و World-Effect Loop corpus

واجهات ومخططات واختبارات مقترحة محدد هندسيًا 3E

نتيجة runtime محفوظة مع trace مجرّب محليًا 4E

مكسب يعاد إنتاجه مقابل baseline مقارن ومكرر 5E

لا ترفع هذه الورقة RFAGI إلى 4E أو 5E. وضعه الحالي 3E: تعريف نظري، معمارية، عقود، وبرنامج اختبار. بعض آليات MOUSSAIF الأصلية تقع أيضا في 2E–3E بحسب المادة المسترجعة. لا يجوز استنتاج deployment أو benchmark من وجود PDF أو كود أو prompt. [M2]

أما المصادر الخارجية فتستخدم كمقارنة لا كإثبات مباشر لصحة RFAGI. نجاح SABER في verification gated-mutation، مثلًا، يدعم أهمية حماية الخطوات المغيرة للحالة، لكنه لا يثبت أن Fabric Reliability ككل سيحقق gain. وبالمثل، وجود rollback frameworks لا يثبت صلاحية معادلاتنا أو machine state المقترحة؛ هذه تحتاج تجاربها الخاصة.

<PARSED TEXT FOR PAGE: 6 / 49>

3. حالة الفن: من success task إلى reliability agent

تتحرك حالة الفن في ثلاثة اتجاهات متقاربة. الاتجاه الأول هو تقييم الوكيل داخل بيئات حقيقية قابلة للتنفيذ. OSWorld يبني 369 مهمة عبر أنظمة تشغيل وتطبيقات فعلية مع evaluator تنفيذي للحالة، وهو يرسخ أن التقييم يجب أن يقرأ البيئة لا أن يطابق جواباً نصيًا فقط.]ToolSandbox[4R يضيف execution stateful واعتماد الحالات بين الأدوات وevaluation milestone خلال trajectory، وهو قريب من فكرة postconditions الجزئية والنهائية.]GAIA[3R من جهته يختبر reasoning والتصفح والملفات والأدوات في مسائل بسيطة للبشر لكنها صعبة للوكلاء.]12R[

الاتجاه الثاني هو consistency والموثوقية عبر runs. -bench τ يقارن state database النهائي بالهدف ويقدم k^pass لقياس الاستقرار عبر محاولات متعددة، انخفاض 8^pass مقارنة بالنجاح المفرد يكشف أن agent قد يبدو جيدًا في المتوسط لكنه غير موثوق عند الاستخدام المتكرر.]2R]ويذهب Rabanser وآخرون أبعد من ذلك بتفكيك reliability إلى consistency وrobustness وpredictability وsafety، مع مقاييس للتباين في المسار والموارد، perturbations prompt/environment/fault، calibration، وشدة الضرر.]1R[

الاتجاه الثالث يحمي الأفعال ذات العواقب. SABER يحلل steps mutating ويجد أن الانحرافات فيها أكثر حسمًا للفشل من الانحرافات غير المغيّرة للحالة، ثم يستخدم verification gated-mutation وReflection Targeted قبل هذه الخطوات.]GA-[7R Rollback يفصل مولّد الفعل عن مساعد يفحص كل خطوة ويطلق rollback عند اكتشاف خطأ.]8R]ويقترح by Revisable taxonomy Design للأفعال: Idempotent، Reversible، Compensable، Irreversible، مع النتيجة المركزية أن مرونة الوكيل محدودة بقابلية تراجع مساحة أفعاله.]9R[

تضيف أعمال الأمن طبقة أخرى. AgentDojo يختبر injection prompt عبر أدوات وبيانات غير موثوقة، بينما Security Agent Bench يغطي طيفًا واسعًا من الهجمات والدفاعات، ويظهر أن الذاكرة واستدعاء الأدوات وسياق المستخدم نقاط هجوم فعلية.]6R5,R[ToolEmu يستخدم sandbox محاكى لاكتشاف مخاطر الأدوات مرتفعة العاقبة قبل تنفيذها في العالم الحقيقي.]11R]وأعمال use tool safe verifiably تحاول الانتقال من safeguards استدلالية إلى متطلبات قابلة للفرض على flows data وتسلسل الأدوات.]R10[

الفجوة التي تستهدفها الورقة

الأدبيات تقدم مقاييس، benchmarks، gates، rollback، وأمنًا، لكن RFAGI يقترح جمع هذه الوظائف في حاكمية واحدة مرتبطة مباشرة بالهدف، MCD، authorization، effect-world closure، recovery، والذاكرة المتطورة. هذه دعوى تصميمية تحتاج اختبارًا وليست ادعاء سبق علمي مثبتًا.

<PARSED TEXT FOR PAGE: 7 / 49>

4. الأسس المباشرة في MOUSSAIF Corpus

لا يبدأ RFAGI من فراغ. تقرير MOUSSAIF / OS AGI MOUSSAIXT يضع خمسة invariants ذات صلة مباشرة: الدستور قبل التحسين، الآلية قبل الخطاب، أصغر فعل كامل قابل للتراجع، «happen not did it or trace»، وBLOCKED/PARTIAL الصادق أفضل من نجاح مختلق. كما يفصل 4T–0T Authorization بين reasoning وبين external side effects، ويشترط verification/stabilization. [M1] حتى حاصال rollback وبقاء mutable high-risk work قبل checkpoint

في Contract Claim & Hierarchy Evidence، يحمل كل ادعاء status ومصدرًا واختبار ترقية، مع تمييز من شكل أخطر منع الغرض. reported/requested/created/file observed/locally verified/benchmarked/deployed.]M2[.دليله من أقوى claim إلى tool response أو artifact تحويل :success inflation

Verified ولا Verified Action، مع Attempted ولا Attempted، مع Proposal يسوى لا World-Effect Loop، في observable رفّتعُ التنفيذ قبل Measured Effect. مع External Result ولا External Result، مع Action invariants-و intended-versus ويفحص authoritative environment قراءة verifier يعيد وبعده postconditions، actual state. [M1]

في Observability، كل يسجل أن يجب run objective، observed state، assumptions، evidence grades selected skills، router scores، plan nodes، tool calls، tests، failures/repairs، world effects، unresolved risks، rollback، وcandidates learning. وفي Recovery Failure، الفشل state لا exception مخفي؛ يصنف ويعالج عبر postconditions تحقق قبل التعافي يكتمل ولا ،bounded retry/alternate/repair/dry-run/partial/block/rollback rollback. [M1,M2] بال الخاصة

وأخيرًا تظهر في QRET« نظرية الموثوقية التصعيدية«: الموثوقية لا تأتي مرة واحدة، بل تصعد من اتساق أولي إلى دليل ثم اعتراض ثم اختبار ثم أثر عالمي.]3M]هذه العبارة تتحول في الورقة الحالية من نظرية معرفية فرعية إلى contract معمارية كاملة للوكالة.

<PARSED TEXT FOR PAGE: 8 / 49>

5. ما الموثوقية؟ تمييزها عن القدرة والسلامة والصلابة

نعرف Capability بأنها مجموعة الأفعال التي يستطيع النظام تنفيذها في شروط معينة. Reliability هي احتمال ونمط أن يعمل النظام كما هو محدد عبر التكرار والاضطراب والزمن. Robustness جزء منها يتعلق بالحفاظ على الأداء عند perturbation. Safety تتعلق بحدود العاقبة والمخالفات، ولا ينبغي دمجها حسابيًا بحيث تخفي tails catastrophic. Verification هي عملية إنتاج دليل على postcondition، وليست مرادفًا للموثوقية. Observability هي قابلية معرفة ما حدث ولماذا. Recoverability هي قدرة العودة إلى حالة مقبولة بعد الفشل. Governance هي النظام الذي يحدد من يملك سلطة الانتقال بين هذه الحالات.

خطأ الخلط الشائع السؤال الذي يجيب عنه المفهوم

اعتبار الإمكان ترخيصًا أو نجاحًا هل يمكنه فعل X؟ Capability

إخفاء التباين وشدة الخطأ كم مرة أعطى نتيجة صحيحة؟ Accuracy

Reliability

هل يعمل كما ينبغي عبر runs/perturbations؟

اختزاله في accuracy

قياس nominal فقط ماذا يحدث عند تغير البيئة أو تعطل أداة؟ Robustness

متوسط score يخفي risk tail ما حدود الضرر عند الفشل؟ Safety

الثقة الذاتية بدل الدليل هل ثبتت postcondition؟ Verification

retry بدل recovery هل يعود إلى حالة آمنة؟ Recoverability

ترك القرار للLLM وحده من يسمح بالانتقال والفعل والتعلم؟ Governance

تعريف RFAGI للموثوقية

كامل لمسار خاصية هي Reliability: + specification + authorization + execution + observation + verification + memory + recovery. لذلك لا يمكن إسنادها إلى النموذج وحده، ولا إلى الأداة وحدها، ولا إلى evaluator وحده.

<PARSED TEXT FOR PAGE: 9 / 49>

6. الأطروحة: Reliability as Governing Infrastructure

الأطروحة الأولى أن الموثوقية ليست feature توضع بعد loop agent، بل plane يملك حق منع الفعل، تخفيضه إلى run dry، فرض verifier مستقل، طلب approval، أو إيقاف التعلم من trajectory. بهذا المعنى هي »حاكمية« لأنها لا تنتج المحتوى الرئيسي للمهمة؛ بل تحدد متى يجوز تحويل التفكير إلى أثر، ومتى يجوز تحويل الأثر إلى حقيقة في الذاكرة.

الأطروحة الثانية أن closure Reliability أهم من correctness local. قد تكون كل خطوة منفردة معقولة ومع ذلك تفشل المهمة بسبب تراكم drift، context stale، سوء ترتيب effects side، أو عدم تحقق شرط أخير. لذلك يجب أن تتبع المهمة state machine صريحة، ولا تترقى إلى SUCCESS_VERIFIED إلا إذا أغلق loop على state authoritative.

الأطروحة الثالثة أن الاستقلالية محكومة بقابلية التراجع. كلما اقترب الفعل من irreversible أو radius-blast-high، ارتفع checkpoint/approval/independent على demand وازداد allowed autonomy، انخفض required evidence، verification. هذه ليست قاعدة أخلاقية فقط؛ بل استراتيجية لتقليل loss expected.

الأطروحة الرابعة أن التعلم نفسه يحتاج Gate Reliability. إذا تعلّم النظام من trajectory ناجحة ظاهرًيا لكنها فشلت في العالم، فهو يبني loop poisoning-self. لذلك الخبرة القابلة للترقية إلى HyperSkill يجب أن تحمل evidence، analysis causal، ونتيجة قابلة لكن« MOUSSAIF: «Every repeated win becomes a candidate HyperSkill قانون يوسع هذا .الاختبار لإعادة promotion requires transfer testing، و»Every failure changes policy or memory». [M5]

× Reliable agency = Capability × Governance × Verification

Recoverability × Verified Learning

<PARSED TEXT FOR PAGE: 10 / 49>

7. ملف الموثوقية: تسعة محاور بدل رقم واحد

تقترح الورقة ملفًا vectorial لأن score واحدًا قد يخفي محورًا منهارًا. لا سيما safety والauthorization يجب أن يبقيا gates hard.

المحاور التسعة هي:

أمثلة Metrics ما يقيسه الرمز المحور

Epistemic Reliability R_epi

صحة المصدر والادعاء وحدود

اليقين

evidence grade, false claim rate

Reasoning Reliability R_reas والحاكمية والمسار القيود ثبات constraint retention, court catch rate

Tool Reliability R_tool

invocation/response سلامة

handling

tool error recovery, schema

accuracy

Execution Reliability R_exec وترتيبها الأفعال صحة mutation error rate, step success

World-Effect Reliability R_world postconditions مع الواقع تطابق closure rate, state gap

Predictability R_pred

معايرة الثقة واكتشاف حدود

النجاح

ECE/Brier/AUROC

Recovery Reliability R_rec الفشل بعد rollback/الإصلاح rollback success, MTTR

جودة ما يترقى إلى الذاكرة mem_R Reliability Memory

contamination rate, provenance

completeness

Governance Reliability R_gov والحدود بالتفويض الالتزام
unauthorized action rate, gate
coverage
يمكن اشتقاق aggregate للاستخدام التشخيصي، لكن deployment لا يجب أن يعتمد عليه وحده. إذا كان world_R مرتفعًا بينما gov_R منخفض، فقد يصل النظام إلى النتائج الصحيحة بطرق غير مأذونة. وإذا كان exec_R مرتفعًا بينما mem_R منخفض، قد ينجح اليوم ويقوّض نفسه غدًا. لذلك يعرض dashboard المحاور منفصلة ويستخدم constraints hard للأمان والتفويض.
<PARSED TEXT FOR PAGE: 11 / 49>

8. النظرية الأولى: نظرية الموثوقية التصعيدية الممتدة
الأصل في QRET يحدد سلسلة: اتساق دليل اعتراض → → → → اختبار أثر عالمي.]3M[توسع الورقة هذه السلسلة إلى سبع R3 executable، R2 adversarial objection، R1 provenance ودليل، coherence داخلي؛ هي R0 الدرجة .تشغيلية درجات لا .runs/perturbations عبر R6 replicated reliability؛ R5 measured world effect؛ R4 verified action، test؛
يجوز أن يقفز claim من 0R إلى 5R بسبب بلاغة الجواب أو ثقة النموذج.
ما يسمح به المعنى الدرجة
اقتراح فقط اتساق لغوي/منطقي أولي 0R
استخدام معرفي منخفض المخاطر مصدر ودليل قابلان للتتبع 1R
تخطيط/محاكاة اعتراضات ومحكمة ولم يظهر مانع حاسم 2R
تشغيل sandbox اختبار تنفيذي محدد مع predicate success 3R
ترقية status إلى action verified الفعل نفسه متحقق على state target 4R
closure effect-world الأثر الخارجي مقاس ومستقر 5R
R6 variations تحت تتكرر النتيجة production/skill promotion مرشح
فرضية النظرية: كلما طالت السلسلة بين 0R و5R دون gates صريحة ارتفع rate success-false. ما يهدمها: إذا لم يخفض تطبيق الدرجات inflation success أو لم يحسن تشخيص الفشل مقارنة بbaseline بسيط، فالتصعيد مجرد overhead اصطلاحي.
<PARSED TEXT FOR PAGE: 12 / 49>

9. النظرية الثانية: Reliability Closure Theory
تعرف Closure Reliability بأن المهمة لا تكتمل عندما يتوقف executor، بل عندما يغلق الفرق بين state desired وobserved authoritative state ضمن tolerance محددة .ف tool response ليس closure، وfile creation ليس closure
إذا كان المطلوب deployment، وacknowledgment API send ليس closure إذا كان المطلوب تسليم رسالة ويجب فحص الحالة أو receipt المناسب.
Closure = Verify(Observed_PostState, Declared_Postconditions,
Invariants, Residual_Gaps)
يبنى closure contract قبل التنفيذ: Preconditions، Success Predicate، Negative Constraints، Allowed Side Effects، Measurement Source، Stabilization Window، Rollback Predicate. بعد التنفيذ يحسب State Diff بين before/after، ويعطي حالات من واحدة :VERIFIED_SUCCESS، VERIFIED_PARTIAL، VERIFIED_WRONG_EFFECT، VERIFIED_NO_EFFECT، UNVERIFIABLE، ROLLED_BACK.
المساهمة المهمة هنا هي فصل »الفعل« عن »الأثر«. قد يكون الفعل صحيحًا لكن الأثر لم يتحقق بسبب eventual consistency أو نظام خارجي، وقد يكون الأثر تحقق جزئيًا مع effect side غير مقصود. لذلك يخزن ledger كلاهما بدل Boolean نجاح واحد.
معيار الهدم
إذا كان verification diff-state لا يلتقط أخطاء إضافية مقارنة بقراءة response tool فقط، أو كانت تكلفة التحقق أعلى من قيمته في المهام المحددة، يجب تقييد النظرية بمجالات impact-high/mutating بدل تعميمها.
<PARSED TEXT FOR PAGE: 13 / 49>

10. الثالثة النظرية: Reversibility-Bounded Autonomy
تنص النظرية على أن مستوى autonomy المسموح يجب أن يتناسب عكسيًا مع irreversibility وradius blast، وطرديًا مع evidence وcoverage verification. يتقاطع ذلك مع 4T–0T في MOUSSAIF ومع taxonomies الحديثة للأفعال القابلة للتراجع أو التعويض. [9R1,M]

Allowed Autonomy ∝ (Evidence × Verification × Recovery) /
(Irreversibility × BlastRadius × Uncertainty)

سياسة RFAGI وصف الفعل Class

Idempotent 0A إعادة التنفيذ لا تغير النتيجة بعد المرة الأولى retries bounded + autonomous

Reversible 1A يمكن استعادة الحالة السابقة مباشرة verify + execute + checkpoint

A2 Compensable

لا يمكن undo حرفيًا لكن يمكن تعويض

compensation plan + verifier + budget الأثر

Irreversible 3A الأثر لا يمكن عكسه عمليًا evidence strongest + approval explicit +
dry run

A4 Prohibited constitution/policy يخالف capability كانت مهما BLOCK

توقع قابل للاختبار: Gate Mutation على 3A–1A سيخفض deviations decisive وشدة الفشل على حساب cost/latency. إذا لم يظهر هذا off-trade أو لم يتحسن severity error، فلا مبرر معماري للبوابة.

<PARSED TEXT FOR PAGE: 14 / 49>

11. الرابعة النظرية: Failure-as-Knowledge Theory
في MOUSSAIF« الفشل أصل معرفي». [2M]توسع الورقة ذلك إلى قاعدة: لا تسأل فقط هل فشل agent؛ بل ما الفرق التشخيصي الذي كشفه الفشل عن model world أو graph capability أو contract tool أو policy. يفترض كل Object Failure أن يحمل category، node، evidence، retry_count، blast_radius، repair_options، rollback_ref، root-cause وconfidence، residue.
يشمل الأولي التصنيف: tool outage، schema mismatch، stale context، insufficient evidence، contradiction، budget exhaustion، no-progress loop، hidden side effect، verification mismatch، permission denial، environment drift، policy conflict، memory inconsistency. لكل نوع recovery policy مختلفة. إعادة المحاولة ليست strategy عامة؛ بل خيار مشروط عندما يكون transient failure وقابلًا للتكرار الآمن.
المعيار الحاسم هو causal patch. إذا حدث failure ثم أعاد النظام المحاولة ونجح دون أن يعرف سبب النجاح، لا يجوز ترقية ذلك مباشرة إلى skill. وإذا تكرر نفس signature failure يجب أن يتغير router أو precondition أو adapter tool أو policy. وإلا تصبح الذاكرة مجرد سجل أخطاء لا نظام تعلم.

Failure Value = Diagnostic Information × Causal Confidence × Reusable
Patch

<PARSED TEXT FOR PAGE: 15 / 49>

12. الخامسة النظرية: Verified Experience Learning
التعلم الذاتي في نظام وكيل يحمل خطرًا بنيويًا: إذا كانت labels تأتي من نجاحات ذاتية غير متحققة، يمكن للنظام أن يضاعف أخطاءه.
تربط VEU كل Verified Experience Unit (VEU). بل trajectory ليست unit of learning أن النظرية تقترح لذلك intent، before-state، action trace، evidence، after-state، verifier verdict، residual gaps، court verdict، وtransfer conditions.
transfer test أو replay ونجحت، minimum evidence rank استوفت إذا إلا HyperSkill Factory إلى VEU تدخل لا عندما يكون الادعاء عامًا. trajectory failure يمكن أن تكون VEU أيضًا، لكن label فيها هو »ما لا يعمل ولماذا« لا »خبرة نجاح«. بهذا تصبح الذاكرة حارسة للموثوقية بدل أن تكون amplifier للخطأ.
corpus: every win must survive a court، promotion في capability federation دستور مع تتوافق الفكرة هذه

consequential action bounded. وكل ،every failure changes policy or memory ،requires transfer testing
[5M الجديد هنا هو جعل schema VEU وسيطًا إجباريًا بين runtime وlearning continual.
فرضية
تدريب/mining-skill من VEU المتحققة سيقلل contamination memory وregression مقارنة بتعدين جميع trajectories الناجحة ذاتيًا. يمكن هدم الفرضية بتجربة learning-continual controlled.
<PARSED TEXT FOR PAGE: 16 / 49>

13. المقترحة المعمارية :L9.5 Reliability Fabric
Reliability الورقة تقترح [M1] .L10 Verification/World Effect ثم L9 Tools/Execution تضع الأصلية المعمارية
Fabric كـplane cutting-cross منطقي يتموضع مفاهيميًا عند 9.5L لكنه يمتد من التخطيط إلى الذاكرة. الغرض ليس إضافة طبقة بيروقراطية، بل تجميع العقود التي تمنع التنفيذ من تجاوز reversibility/authorization/evidence والذاكرة من تجاوز verification.
المسار المعماري
L0 Constitution L1 Goal L2 MCD/QMCD/MCAI L → → → → 3–L8 Knowledge/Skills/Agents L9
Execution L → → → 9.5 Reliability Fabric L10 Verification & World Effect L11 Memory/Evolution
الأعلى من Fabric تتلقى: mission intent، risk score، evidence state، authorization tier، expected postconditions، selected plan. الأسفل من وتتلقى :tool receipts، raw observations، state deltas، errors timing، costs. تقرر ثم :execute، require verification، checkpoint، downgrade، retry، alternate، rollback block، أو promote experience.
المبدأ التصميمي هو separability: يجب أن يستطيع الباحث تعطيل كل وحدة وقياس أثرها. إذا كانت Fabric صندوقًا أسود كبيرًا لا يمكن ablate مكوناته، فستصبح مستحيلة التقييم. لذلك تقسم إلى modules صغيرة بعقود readable-machine.
<PARSED TEXT FOR PAGE: 17 / 49>

14. مكونات Fabric Reliability — العقود قبل الفعل
المخرج المدخلات الوظيفة المكّون
Risk/MCD Adapter
تحويل أثر/غموض/لا رجعة إلى
reliability level
MCD, uncertainty, blast radius RAL 0–4
Authorization Gate permission عن capability فصل action, target, tier
/ALLOW/DRY_RUN
APPROVAL/BLOCK
Reversibility Classifier A0–A4 تصنيف tool contract, target reversibility class
Precondition Checker صالحة البداية نقطة أن إثبات PASS/FAIL + evidence world state, assumptions
Checkpoint Manager last safe state حفظ target state checkpoint_ref
Postcondition Compiler
تحويل الهدف إلى predicates
قابلة للرصد
mission goal, constraints success/negative predicates
Mutation Gate
وقف step mutating حتى
استيفاء الشروط
action + evidence + predicates execute token
Gate Mutation هو قلب reliability action-pre. لا يستدعي »reflection »عامة لكل خطوة؛ لأن ذلك قد يرفع التكلفة دون فائدة. بل يفعل verification targeted عندما تكون الخطوة مغيرة للحالة أو عالية MCD. هذا يتقاطع مع نتائج SABER التي تشير إلى أن

deviations في actions mutating أكثر حسًما للفشل.]7R[
Checker Precondition يمنع خطأ execution state-stale: إذا ُبني plan على حالة قديمة، يجب إعادة القراءة قبل step mutating.
git commit، transaction savepoint، version ID، يكون قد كامل، snapshot مأدائ يعني لا Checkpoint Manager copied artifact، أو logical compensation reference.

<PARSED TEXT FOR PAGE: 18 / 49>

15. مكونات Fabric Reliability — بعد الفعل والتعافي
المخرج الرئيسي الوظيفة المكّون
Receipt Action تسجيل ما طلب وما نفذ فعلًيا response + parameters + id_call_tool
Verifier Independent إعادة قراءة target من مصدر مستقل نسبًيا verdict verification
State Diff Engine before/intended/after مقارنة delta + residual gap
Scanner Effect-Side البحث عن تغييرات غير مقصودة changes unexpected
Failure Classifier Failure Object إلى الخطأ تحويل failure_type + confidence
Recovery Manager
اختيار
retry/alternate/repair/rollback/escalate
recovery plan
Verifier Rollback إثبات العودة أو التعويض closure rollback
Evidence Ledger والحالة بالدليل claim ربط provenance graph
Verified Experience Gate verdict قبل التعلم منع reject / VEU
Reliability Memory والأنماط residuals حفظ reconstructable memory patch
استقلال verifier نسبي وليس مطلقًا. إذا كانت API action وAPI verification نفس المصدر ونفس mode failure، فقد نحصل على correlated false positive. تسجل لذلك Fabric درجة استقلال، evidence: same-call، same-service new read، endpoint alternate، system independent، observation physical/human. كلما ارتفع الأثر يجب أن ترتفع استقلالية التحقق الممكنة.
Scanner Effect-Side مهم لأن النجاح المقصود قد يصاحبه ضرر غير مقصود. لذلك predicate success وحده غير كاف ؛ نحتاج constraints negative مثل »لم يحذف سجلات أخرى«، »لم يرسل إلى recipients إضافيين«، »لم يغّير permissions«،« لم يسّرب secret«.

<PARSED TEXT FOR PAGE: 19 / 49>

16. Reliability State Machine
الحالات
PROPOSED EVIDE → → → → → NCE_READY AUTHORIZED CHECKPOINTED ATTEMPTED
ACTION_OBSERVED → → → POSTCONDITION_VERIFIED WORLD_EFFECT_MEASURED
STABILIZED LEAR → NABLE
ًء يمنع machine state القفز الدلالي. PROPOSED يعني أن agent يملك خطة فقط. ATTEMPTED حدث يعني أن استدعا
لكن ليس أن target تغير. OBSERVED_ACTION يعني أن الأداة أو report environment أعطى evidence أوليًا.
VERIFIED_POSTCONDITION يعني أن الحالة authoritative تطابق predicate.
MEASURED_EFFECT_WORLD يعني أن الغاية العملية الأوسع قيَ ست، مثل وصول الملف إلى المستودع الصحيح أو انخفاض خطأ النظام بعد fix. STABILIZED يضيف نافذة زمنية عندما يكون الأثر eventual أو قابًلا للارتداد. LEARNABLE تعني أن trajectory صارت صالحة للذاكرة/mining skill.
الفروع السلبية مهمة بنفس قدر الفرع الناجح: POLICY_BLOCKED، PERMISSION_BLOCKED، PARTIAL_EFFECT، VERIFICATION_MISMATCH، FAILED_ACTION، FAILED_PRECONDITION،

،UNEXPECTED_SIDE_EFFECT، RECOVERY_REQUIRED، ROLLED_BACK، COMPENSATED
UNVERIFIABLE. هذه الحالات تمنع واجهة المستخدم والسجل الداخلي من القول »تم« عندما لا يوجد closure.
∧ VERIFIED_SUCCESS requires: Authorization ∧ ActionEvidence
PostconditionClosure ∧ NoCriticalSideEffect
<PARSED TEXT FOR PAGE: 20 / 49>

17. التكامل مع QMCD/MCD: ميزانية الموثوقية المتكيفة
MCD في corpus يوجه العمق والموارد إلى الفروق الأعلى قيمة.]1M[تقترح الورقة استعماله أيضًا لتحديد Reliability
RAL (Level Assurance). بدلًا من تشغيل أقصى تحقق لكل مهمة، يرفع النظام الحماية عندما يزيد impact، uncertainty،
أو dependency depth، evidence conflict، irreversibility.
Risk* = w1·Impact + w2·Uncertainty + w3·Irreversibility +
w4·EvidenceConflict + w5·BlastRadius
الضمانات متى RAL
RAL-0 read-only / low impact basic provenance + output check
RAL-1 reversible local precondition + postcondition read
RAL-2 mutating medium impact checkpoint + mutation gate + state diff
RAL-3 high impact / third party
+ independent verifier + approval
rollback/compensation
RAL-4 critical / irreversible
+ strongest evidence + simulation
human/policy gate + dual verification
QMCD يضيف السؤال الحاكم: قبل إنفاق budget verification يسأل »ما الاختبار الأقل تكلفة الذي يفصل بين النجاح الوهمي والنجاح الحقيقي؟«. بهذه الطريقة لا تصبح reliability مرادفًا للبطء، بل توّجه القياس إلى الفرق الذي يغير الحكم.
<PARSED TEXT FOR PAGE: 21 / 49>

18. التكامل مع MCD-HAKIM وMCAI والمحاكم
يعمل MCD-HAKIM في المشروع كمسار تحويل السؤال والفرق إلى pipeline حاكمي. Fabric Reliability لا يستبدله؛ بل يقدم له contracts تنفيذية. قبل المحكمة: Ledger Evidence وContract Claim يحددان الرتبة. أثناء المحكمة:
agents counterexample/verifier يحاولون كسر plan. بعد المحكمة: Gate Mutation يفرض verdict على التنفيذ بدل أن يبقى نصًا استشاريًا.
MCAI مهم عندما توجد حاكميات جزئية: planner يريد completion، security يريد الحد، router cost يريد تقليل الإنفاق، intent user يريد السرعة، verifier يريد evidence أقوى. governance-Meta لا يعني جمع الأصوات؛ بل ترتيب الأسبقية.
authorization/Constitution لا يجوز أن يخسر أمام score optimization.
تقترح الورقة Court Reliability مكونة من أدوار لا يلزم أن تكون agents منفصلة دائمًا: Mutation، Judge Evidence، model أو rules أو kfunctions تنفيذها يمكن. Calibration Judge، Recovery Judge، Side-Effect Judge، calls بحسب RAL. معيار التصميم هو أقل تركيب يحقق الفرق، لا أكبر mesh agent.
anti-inflation قاعدة
لا ننشئ Court أو Agent جديدًا إذا كان deterministic rule أو verifier بسيط يحقق نفس التمييز بتكلفة أقل. هذا متسق مع قانون -3vNEXT: لا تنشئ ما لا يقدم difference operational قابلًا للقياس.]4M[
<PARSED TEXT FOR PAGE: 22 / 49>

19. تصنيف الفشل: من Error إلى Object Diagnostic

المعالجة الافتراضية مثال الفئة
استرجاع/خفض claim/طلب توضيح مصدر غير كافٍ أو متناقض Evidence 1F
F2 Planning خاطئ ترتيب أو missing dependency replan from last safe milestone
bounded retry ثم F3 Tool timeout/schema/error alternate adapter
F4 State stale context / target changed refresh + revalidate preconditions
F5 Permission authorization نقص أو رفض downgrade/block/approval
F6 Verification tool says success but state differs repair/rollback + incident
F7 Side Effect مقصود غير تغيير contain + rollback/compensate
F8 Loop no-progress / repeated identical error stop + escalate
F9 Budget cost/time/context exhausted partial with explicit residue
F10 Memory متعارضة canonical state أو خبرة quarantine + provenance audit
F11 Security prompt injection / poisoned tool data isolate + policy enforcement
F12 Environment API/UI semantics drift adapter update + regression suite
التصنيف نفسه يحتاج calibration؛ لا يجب أن يدعي cause root كاملًا من أول observation. لذلك يحمل Object Failure confidence وcauses alternative، ويمكن تحديثه مع evidence جديد. هذا يمنع policy repair من التصلب حول تشخيص خاطئ.
<PARSED TEXT FOR PAGE: 23 / 49>
MOUSSAIF AGI OS / MOUSSAIXT — Reliability-First AGI
23 | ورقة بحثية هندسية 2026 — Moussaif Zouhair / زهير مساعف
20. Engineering Recovery: لماذا retry ليس تعافيًا
التعافي يبدأ من سؤال: هل سبب الفشل transient أم structural أم epistemic أم authorization؟ retry مفيد فقط في loop بل resilience ليس schema mismatch بعد inputs نفس مع call نفس تكرار. budget وبحدود transient faults يستخدم لذلك. ضائع Recovery Manager سياسة: → → classify snapshot choose retry/alternate/repair/rollback/escalate verify → → write residue، الموحد الكتاب في موجودة صياغة وهي.]M2[
نضيف مفهوم Closure Recovery: كما أن نجاح الفعل يحتاج postcondition، نجاح rollback يحتاج postcondition أيضًا. إذا كان API rollback قال OK لكن state لم يعد إلى baseline، لا يحق للنظام إغلاق incident. compensation بدورها لا تعني إعادة الماضي؛ بل بناء state جديدة مقبولة ومثبتة.
∨ Recovery Success = SafeStateRestored
AcceptableCompensationVerified
metrics: rollback success rate، compensation success، mean time to recovery (MTTR)، تسجل repeated-error rate، residue severity، percentage of failures converted into tested patches. صفر ليس الهدف. فشل - وهو غير واقعي في أنظمة مفتوحة - بل فشل مرئي، محدود، قابل للتشخيص والاسترجاع.
<PARSED TEXT FOR PAGE: 24 / 49>
MOUSSAIF AGI OS / MOUSSAIXT — Reliability-First AGI
24 | ورقة بحثية هندسية 2026 — Moussaif Zouhair / زهير مساعف
21. Observability كشرط للموثوقية لا كلوغ إضافي
»في MOUSSAIF قاعدة هندسية لا شعار.]2M1,M[لكي يكون trace مفيدًا، يجب أن happen not did it or trace« :الأدنى الحد. لاًهائ صاًن يجمع أن لا causal chain يربط run_id، mission_id، state_before_hash، active_difference، selected_policy، plan_node، tool_call_id، action class، evidence refs، state delta، verifier verdict، failure object، recovery action، state_after_hash، cost/latency، residual risk.
يدعم هذا السجل خمسة استخدامات: audit يشرح لماذا أصبحت الحالة SUCCESS_VERIFIED؛ replay يعيد trajectory؛ regression يكشف كسًرا بعد تحديث؛ mining skill يستخرج أنماًطا ناجحة؛ regret-router يقارن قرار routing بما كان يمكن أن يكون أفضل. إذا تعذر استخدام logs لهذه الأغراض، فهي telemetry غير حاكمة.
المعنى Metric
Trace Completeness event + evidence تحمل التي الحرجة transitions نسبة
Verification Coverage postcondition check تملك التي mutating/high-impact steps نسبة
Linkage Evidence نسبة claims المرتبطة بمصدر/اختبار

نسبة runs التي يمكن إعادة بنائها من السجل Replayability
زمن الوصول من verdict إلى أسبابه Latency Audit
قدرة كشف تعديل أو فقد trace Detection Tamper
<PARSED TEXT FOR PAGE: 25 / 49>

Claim Contractو 22. Evidence Ledger
كل claim في runtime يجب أن يحمل claim_id، text/semantic assertion، source، status، scope، validity window، test، evidence rank، and promotion condition. هذا فصل مباشرة يطبق corpus بين created وobserved deployed. [M2] وbenchmarked وlocally verified
Evidence Ledger ليس bibliography فقط؛ هو graph بين claims والأفعال والحالات. مثال: claim« الملف نُشر« يحتاج claim وبالمثل false promotion. هو بينهما الخلط local file evidence. يكفيه« أنشئُ الملف »claim، deployment evidence »الرسالة أرسلت« يختلف عن »تم إنشاء draft »وعن »تم قبول request send »وعن »تم تسليمها« حسب ما يسمح به النظام الخارجي من evidence.
قاعدة الترقية
لا يُرقى status إلا بدليل من نفس رتبة الادعاء أو أعلى. Evidence من رتبة creation لا يثبت deployment؛ confidence لا يعوض غياب evidence؛ والذاكرة لا تُعامل كـstate authoritative إذا كان target قابلًا للقراءة.
تُستخدم window validity لأن reliability زمنية. status صحيح بالأمس قد يصبح stale اليوم. لذلك كل claim مرتبط بـpolicy freshness وtrigger revalidation عندما تكون العاقبة عالية.
<PARSED TEXT FOR PAGE: 26 / 49>

23. نموذج قياس رسمي: Gates + Profile
نستفيد من Rabanser et al. في فصل safety/predictability/robustness/consistency، لكن نضيف محاور مرتبطة بالعقود التنفيذية في [MOUSSAIF.] 1R لتجنب إخفاء المخاطر، نستخدم Profile Reliability بدلًا من average واحد. يمكن حساب Core Reliability هندسيًا لأغراض المقارنة فقط باستخدام mean geometric حتى يعاقب انهيار أي محور:
R_core = (R_epi · R_tool · R_exec · R_world · R_pred · R_rec · R_mem)^(1/7)
ثم نطبق gates hard: auth_G للتفويض، safety_G للقيود الحرجة، trace_G لاكتمال الحد الأدنى من الدليل. Deployment Reliability ليس »درجة أمان« بل قرار promotion:
Deployable = (G_auth = 1) ∧ (G_safety = 1) ∧ (G_trace = 1) ∧ (R_core ≥ θ_domain)
هذه الصيغة لا تدعي أنها معيار عالمي. θ_domain يجب معايرته حسب المجال. كما أن بعض المجالات ستطلب limits على worst-harm case أو mutation unsafe لا يمكن تعويضها بارتفاع بقية المحاور. لذلك safety خارج المتوسط وُتراجع منفصلة، انسجامًا مع أدبيات reliability الحديثة. [1R]
<PARSED TEXT FOR PAGE: 27 / 49>

24. حزمة Metrics المقترحة
الأفضل التعريف التشغيلي Metric
↑ نسبة المهام المغلقة على الهدف Success Task
↑ محاولات k عبر النجاح pass^k / repeat success
↑ ثبات النتيجة تحت نفس الشروط Consistency Outcome
↑ نسبة الأداء بعد paraphrase Robustness Prompt
↑ تحت الأداء Fault Robustness tool faults
↓ عند كاذب VERIFIED_SUCCESS False-Success Rate audit

↓ فشل لم يعلنه النظام Silent Failure Rate
↓ mutating steps المخالفة لل Unsafe Mutation Rate gate
↑ Verification Coverage postcondition checks / required steps
↓ World-Effect Gap observed state و intended بين المسافة
↑ استعادة/تعويض متحقق Success Rollback
↓ متوسط زمن التعافي MTTR
↓ فجوة Calibration Error confidence/empirical outcome
↑ claims ذات Evidence Completeness provenance/test
↓ خبرات غير متحققة تمت ترقيتها Contamination Memory
↓ تكلفة gain بقاء مع Reliability Overhead reliability / baseline cost
المؤشر الأخطر هنا Rate Success-False لأنه يمثل فجوة operational-epistemic: النظام يعتقد أو يصرح أنه نجح بينما الواقع لا يدعم ذلك. الهدف الأول لـFabric Reliability هو خفض هذا المؤشر حتى لو لم ترتفع success task raw فورا؛ لأن الاعتراف الدقيق بالفشل شرط لاحق لتحسين القدرة.

<PARSED TEXT FOR PAGE: 28 / 49>

.25 البرنامج التجريبي: 12 تجربة أساسية

الفرضية التجربة	ID
R-01 Repeated-run consistency	
Fabric تخفض variance دون خفض success بصورة كبيرة	
R-02 Prompt paraphrase robustness	
حساسية يقلل goal/postcondition compiler الصياغة	
R-03 Tool fault injection fault robustness يرفع Recovery Manager	
R-04 Environment/schema drift	
يقللان precondition refresh + adapter checks brittleness	
R-05 Mutation-gate ablation	
decisive mutating يخفض Mutation Gate errors	
R-06 Postcondition closure false-success rate يخفض state reread	
R-07 Rollback/compensation safe-state restoration يرفع recovery closure	
النظام يفشل closed ولا يتحايل R08- denial Authorization	
R-09 Prompt injection/tool poisoning policy تتجاوز لا untrusted observations	
R-10 Stale-context challenge قديم world model على action يمنع refresh gate	
R-11 Verified-memory learning contamination/regression تقلل VEU	
R-12 Cross-model continuation	
memory + trace يسمحان باستئناف موثوق عبر models/runtimes	

كل تجربة تقارن scaffold agent baseline مع +Fabric ثم ablation للوحدة المعنية. لا يكفي قبل/بعد عام، لأن تحسّنا قد يأتي من mechanism تبديل ثم ،task set ،budgets ،tools ،الأساسية prompts ،model تجميد يجب .أقوى model أو أطول context واحد قدر الإمكان.

<PARSED TEXT FOR PAGE: 29 / 49>

26. مصفوفة Benchmarks الخارجية والداخلية

كيف تُستخدم في RFAGI ما تختبره البيئة

τ-bench

+ tool-agent-user + database state

repeatability

consistency، policy following، world-state

closure

ToolSandbox stateful tool use + milestones state dependencies، intermediate verification

OSWorld real computer environment execution-based postconditions + side effects

GAIA browsing/files/multi-step reasoning general coordination + evidence

prompt injection security/governance reliability عبر AgentDojo tool data

Agent Security Bench attacks across agent pipeline memory/tool/prompt robustness

SWE-Bench Verified code mutation + tests mutation gates، rollback، test closure

MOUSSAIF R-Suite project-specific controlled tasks

MCD routing، trace، VEU، cross-kernel

continuation

لا نقترح استبدال benchmarks العامة. المطلوب هو طبقة reliability فوقها: تنفيذ كل task عدة مرات، perturbation، fault injection، severity labels، trace inspection، وقياس rollback/false-success. أي benchmark يعطي success فقط يمكن توسيعه بprofile، مع الحذر من تغيير semantics الأصلية.

<PARSED TEXT FOR PAGE: 30 / 49>

27. محكمة الهدم وAblation

أقوى اعتراض على RFAGI أنه قد يعيد تسمية مبادئ معروفة في distributed systems، SRE، safety engineering، transactional workflows، وevaluation agent دون مكسب جديد. لا يُرد على هذا الاعتراض لغويًا. المطلوب أن تظهر Fabric مكسبًا تفاضليًا reproducible في metrics محددة أو أن تُخفض إلى مجرد pattern integration.

• احذف Verifier Independent: هل يرتفع success-false؟

• احذف Gate Mutation: هل ترتفع الأخطاء الحاسمة في steps mutating؟

• احذف Classifier Reversibility: هل ترتفع شدة الفشل أو تنخفض recovery؟

• احذف Failure Classifier: هل يزيد repeated-identical-error؟

• احذف Verified Experience Gate: هل ترتفع memory contamination/regression؟

• استبدل RAL based-MCD بمستوى ثابت: هل تنخفض الكلفة دون خسارة reliability أم العكس؟

• استبدل reread state بثقة النموذج: ما معدل inflation success؟

محكمة أخرى هي court cost: إذا تضاعفت cost/latency 3 ×للحصول على gain طفيف في مجال منخفض المخاطر، ينبغي تفعيل Fabric فلسفة تنص كما« useful verified capability / (complexity + compute + risk + redundancy)» هدفنا .ياًانتقائ

-3vNEXT، لا أقصى عدد من الوحدات.]4M[

<PARSED TEXT FOR PAGE: 31 / 49>

28. دراسة حالة :1 تعديل ملف أو Artifact

المهمة: »عدّل ملف الإعداد ثم تأكد أن التطبيق يستخدم القيمة الجديدة«. agent baseline قد يكتب الملف ويرد »تم«. RFAGI يفصل المهمة: يقرأ version/file authoritative؛ يحدد predicate success ليس »succeeded save »بل »القيمة الجديدة موجودة ويمكن للتطبيق قراءتها«؛ يصنف الفعل reversible 1A؛ ينشئ checkpoint؛ ينفذ mutation؛ يعيد قراءة الملف؛ يشغل test أو read config من التطبيق؛ يفحص effects side؛ ثم يغلق state.

Trace مختصر

PROPOSED RAL-2 AUTHORIZED checkpoint → → → → → → =v17 write file reread PASS app config

probe PASS diff only → → → _expected_key VERIFIED_SUCCESS VEU candidate
إذا فشل probe app رغم أن reread file صحيح، فالنتيجة PARTIAL_VERIFIED لا success. Object Failure يشير إلى syntax error، في التعديل تسبب إذا correct target file. أو reload service مثل repair يختار ثم، propagation/reload gap ثم rollback verification. هذه الدراسة البسيطة تكشف الفرق بين reliability operation file و effect system reliability.

<PARSED TEXT FOR PAGE: 32 / 49>

29. دراسة حالة 2: معاملة ويب ذات أثر خارجي

المهمة: «غيّر حجزا إلى موعد جديد». هذه فئة أعلى لأن الأثر يخص طرفًا خارجيًا وقد يترتب عليه مال أو التزام. يُصنف 3RAL- وربما تشمل Postconditions. authorization، fees، availability، policy، الحالي الحجز يقرأ :الفعل قبل .A2 Compensable id_booking نفسه أو المرجع الصحيح، الموعد الجديد، عدم وجود duplicate، وعدم خصم غير متوقع.

Gate Mutation يمنع الضغط على confirm إذا كانت الرسوم أو الشروط تغيرت منذ planning. بعد submit لا تكفي صفحة «success»؛ يعاد فتح state booking من endpoint أو UI مستقل. إذا وقع duplicate أو fee غير متوقع، recovery يختار cancellation/compensation حسب policy. التعلم من هذه trajectory لا يتم حتى تستقر الحالة.

تلتقي هذه الحالة مع τ-bench الذي يقيم state database النهائي ومع أدبيات [safeguards mutating.] 7R2,R لكنها تضيف VEU. وrecovery closure وnegative postconditions.

<PARSED TEXT FOR PAGE: 33 / 49>

30. دراسة حالة 3: Agent Coding ونشر Production

المهمة: إصلاح bug ثم نشره. capability raw قد تكون كتابة patch يمر tests unit. profile Reliability يوسع المطلوب: code review policy، محدود؛ diff، regression suite، fix بعد tests، fix قبل issue reproduced أن evidence deployment authorization، canary أو staged rollout، production health postconditions، rollback ref، ونافذة stabilization.

النجاح له مستويان: Verified Code وVerified Effect-World. الأول يعني patch + tests. الثاني يعني أن production behavior تحسن دون regressions حرجة. هذا يمنع الخلط بين «artifact generated» و«artifact deployed»، وهو من MOUSSAIF. [M1] في الأصلية invariants

إذا مرت tests وفشل canary، لا تُعاد كتابة الذاكرة بأن skill ناجحة. تُسجل failure VEU مع specific-environment difference. إذا rollback أعاد metrics health إلى baseline، successful recovery رغم أن المهمة الأصلية فشلت. هذه الدقة تمنع التعلم من narratives only-success.

<PARSED TEXT FOR PAGE: 34 / 49>

31. دراسة حالة 4: موثوقية البحث والادعاء

الموثوقية ليست فقط للأفعال المادية. في البحث، mutation قد تكون ترقية claim في corpus. مثال: «الميكانيزم X ثبت أنه يحسن benchmark». Fabric Reliability المعرفية تسأل: هل لدينا artifact؟ تجربة؟ frozen baseline؟ replication؟ أم مجرد خطة؟

مناسب evidence دون E4/E5 إلى E2/E3 من الترقية يمنع Claim Contract.

تستخدم المحكمة Independence Source، directness، recency، reproducibility، وscope claim. إذا مصدر واحد يذكر النتيجة، لا يجوز توسيعها إلى «حقيقة عامة». إذا paper خارجي يدعم mechanism مشابهًا، يوصف ذلك support comparative لا proof للمكّون المحلي. هذا مهم في مشروع بحثي غني بالتسميات: anti-component-inflation law يطلب operational difference وbenchmark وablation قبل إنشاء أو ترقية مكّون. [4M]

Reliability في البحث

الادعاء الموثوق هو ادعاء يعرف رتبته وحدوده وشروط ترقيته وهدمه، لا ادعاء يملك اسمًا قويًا فقط.

<PARSED TEXT FOR PAGE: 35 / 49>

32. Reliability Security: عندما تصبح Observation نفسها خصمًا
AgentDojo وASB يبرزان أن data المسترجعة من tools قد تحمل injection prompt أو content مصممًا لاختطاف agent. [6R5,R] لذلك Fabric Reliability يجب أن تفرق بين Observation وInstruction. data tool لا تملك سلطة تغيير constitution أو authorization. كل يمر tool output عبر trust label وsource capability contract.
أسوار ثلاثة الورقة تقترح: Data/Instruction Separation، Capability Labels، وMutation Policy Enforcement خارج نص النموذج قدر الإمكان. في -3/4RAL، لا يكفي أن »يتذكر« النموذج عدم تسريب البيانات؛ يجب أن تمنع tool layer policy sequence غير المسموح أو تتطلب approval. هذا ينسجم مع اتجاه use tool safe verifiably الذي يحاول صياغة hazards ومتطلبات قابلة للفرض. [10R]
كما يجب فحص poisoning memory. كل patch memory قادم من observation tool/web يحتاج provenance، ولا يحد من Verified Experience Gate self-reinforcing injection عبر sessions. تكراره بمجرد canonical rule إلى يترقى.

33. Predictability: أن يعرف النظام حدود نجاحه
الوكيل الذي ينجح 90% لكنه يعطي confidence 99% دائمًا قد يكون أخطر من agent ينجح 80% لكنه يميز حالات الشك ويطلب verification. Predictability تسمح بـthresholds: تحت evidence/confidence معينة، يختار النظام run dry أو escalation بدل action.
لكن confidence reported-self وحدها ضعيفة. نقترح Confidence Operational مشتقة من عوامل observable: evidence grade، tool health، state freshness، plan novelty، prior task-family success، verifier independence، contradiction count. يمكن معايرتها ثم دمجها مع model confidence empirically.

```
C_oper = Calibrate(f(Evidence, Freshness, ToolHealth, Novelty,
((Contradictions, PriorSuccess
```

تُقاس calibration عبر AUROC/ECE/Brier كما في الأدبيات الحديثة. [1R] والأهم أن threshold يجب أن يربط بالعاقبة: confidence 0.7 قد يكفي لبحث only-read، لكنه لا يكفي لتحويل مالي أو deletion.

34. اقتصاد الموثوقية: تكلفة الدليل مقابل تكلفة الفشل
لا يمكن جعل كل step تمر بأقصى verification، وإلا يفقد agent قيمة autonomy. لذلك نعامل reliability كميزانية.
Value Verification Expected يزداد مع احتمال الفشل × شدة الأثر × قابلية اكتشاف الخطأ بالاختبار، وينخفض مع تكلفة الاختبار. هذا يجعل RAL/MCD أداة scheduling للثقة.

```
EVV ≈ P(failure) × Severity × DetectionGain − VerificationCost
```

إذا EVV موجب بوضوح، نضيف verifier أو test. إذا سلبي في خطوة only-read منخفضة الأثر، نستخدم sampling أو logging فقط. في production يمكن تطبيق sampling adaptive للمهام المستقرة مع verification full عند anomalies أو model/tool updates.
Metric Reliability Overhead يجب أن يعرض latency، token/API cost، عدد calls، وhuman approval burden.
الهدف ليس أعلى reliability مجردة، بل frontier Pareto بين capability، reliability، cost، وtime. هذا مهم لتفادي نظام »موثوق« لأنه لا يفعل شيئًا.

35. من Trace إلى HyperSkill: سلسلة الترقية الموثوقة
بعد SUCCESS_VERIFIED لا يصبح المسار HyperSkill فورًا. تقترح الورقة سلسلة: xtraction → → e pattern VEU → → → → → → semantic dedup candidate skill replay transfer test ablation benchmark promotion. هذه السلسلة تعكس فلسفة Lab Evolution في candidate -3:vNEXT لا يرّوج نفسه، ويمر

sandbox/replay/benchmark/ablation/security/canary. [M4]
reliability contract: supported intents، preconditions، tools، known failure modes، تحمل نفسها المهارة
success predicate، rollback strategy، evidence rank، test suite، last verified version. عند drift أو failures
متكررة يمكن deprecate/downgrade بدل إبقائها »مهارة« بالاسم.
هذا يربط reliability بالتطور: النظام يصبح أذكى ليس لأنه يراكم skills، بل لأنه يزيد نسبة capabilities التي لها دليل نقل وتكرار وحدود فشل معروفة.
<PARSED TEXT FOR PAGE: 39 / 49>
MOUSSAIF AGI OS / MOUSSAIXT — Reliability-First AGI
39 | ورقة بحثية هندسية 2026 — Moussaif Zouhair / زهير مساعف
36. نموذج نضج RFAGI
شروط الترقية الوصف المستوى
لا يعد موثوقًا agent ينتج/ينفذ دون closure Fluent 0L
L1 Traceable أولية وأدلة trace له run كل trace completeness threshold
L2 Verifiable الحرجة للعمليات state diff وpostconditions منخفض false-success
L3 Recoverable
checkpoint/rollback/compensation
متحققة
recovery tests pass
L4 Predictable calibration + perturbation profile reliability metrics stable
L5 Replicated seeds/tasks/models عبر تتكرر gains E5 evidence
L6 Evolvable
contamination دون verified learning
ملحوظ
continual regression gate
هذا النموذج لا يساوي AGI of Levels ولا يدعي درجة ذكاء. إنه maturity للموثوقية الهندسية. يمكن لنظام محدود capability أن يكون 4L reliability داخل نطاق ضيق، بينما agent واسع capability قد يبقى 1L.
<PARSED TEXT FOR PAGE: 40 / 49>
MOUSSAIF AGI OS / MOUSSAIXT — Reliability-First AGI
40 | ورقة بحثية هندسية 2026 — Moussaif Zouhair / زهير مساعف
37. Blueprint تنفيذي لـMOUSSAIXT
يمكن تنفيذ 2RAGI كخدمات صغيرة أو library داخل kernel. الحد الأدنى، readable-machine: objects MissionSpec، VEU. ActionSpec، PostconditionSpec، EvidenceItem، FailureObject، RecoveryPlan، VerificationVerdict
لا يحتاج الإصدار الأول models إضافية لكل شيء؛ كثير من deterministic gates.
Pseudo-flow
compile mission 2 → → → →) score RAL 3) check authorization 4) classify reversibility 5) compile)1
(pre/postconditions 6 → → → →) checkpoint if needed 7) execute bounded action 8) capture receipt 9
(reread authoritative state 10 → → → →) diff + side-effect scan 11) verdict 12) recover if needed 13
write evidence ledger 14 →) create/reject VEU.
مقترحة واجهات: reliability.prepare(action_spec)، reliability.authorize(context)،
reliability.checkpoint(target)، reliability.verify(postconditions)، reliability.recover(failure)،
أن يجب نفسها Reliability layer. failure mode وinterface timeout لكل reliability.promote_experience(trace).
تكون observable وقابلة للفشل؛ لا نفترض أنها oracle.
<PARSED TEXT FOR PAGE: 41 / 49>
MOUSSAIF AGI OS / MOUSSAIXT — Reliability-First AGI
41 | ورقة بحثية هندسية 2026 — Moussaif Zouhair / زهير مساعف
38. الحدود والمخاطر

• لا توجد في المادة المتاحة نتائج benchmark تثبت أن RFAGI يتفوق على baselines؛ الوضع الحالي proposal 3E.
• التحقق قد يشارك action نفس مصدر الخطأ؛ «verifier independent» استقلال نسبي يحتاج قياسًا.
• Postconditions السيئة تنتج verification صحيحًا لمواصفة خاطئة؛ reliability specification تسبق reliability execution.
• RAL adaptive يلزم ؛user experience وتقلل latency/cost ترفع قد gates زيادة.
• Rollback قد يكون مستحيلًا أو ناقصًا في العالم المادي والاجتماعي؛ compensation لا يعيد الماضي.
• Calibration تتغير مع versions tool/model/domain؛ لا يوجد threshold دائم.
• Benchmarks قد تحتوي أخطاء أو ambiguities؛ SABER نفسه أشار إلى مشاكل في بعض τ-bench tasks.
• Security adaptive adversaries؛ نجاح الدفاع الحالي لا يضمن الصلابة المستقبلية.
• Reliability لا تعادل alignment أو truth أو ethics؛ هي بنية operational dependability ضمن أهداف وقيود محددة.
• المعمارية قد تصبح engineered-over؛ ablation وcourt cost إلزاميان.
الحد الفلسفي الأهم: reliability دائمًا نسبة إلى specification ومجال تشغيل. نظام ينفذ المواصفة الخاطئة بدقة عالية قد يكون reliable تقنيًا لكنه غير صالح غائيًا. لذلك يظل Question Governing/Goal وConstitution أعلى من Fabric Reliability.
<PARSED TEXT FOR PAGE: 42 / 49>

39. الفرضيات والتنبؤات القابلة للاختبار
التنبؤ H
H1
من أكثر false-success يخفض mutating actions بعد state reread
وحدها self-reflection.
H2
من cost لكل أكبر gain يعطي mutation-gated verification
لكل verification step.
من أفضل Pareto يحقق المتكيف RAL. H3 fixed maximum verification.
H4
حتى إذا لم يرفع task يخفض severity classification reversibility
success.
يقلل H5 .repeated-identical-error rate failure classification
H6
بالـ مقارنة continual-learning regression يخفض VEU gating raw
trajectory mining.
مع يزداد independent verification gain وH7 .impact tool uncertainty
بسرعة يتنبأ root-cause analysis وH8 .MTTR trace completeness
H9
يرتبط بثقة المستخدم/المشغل أكثر من raw closure effect-world
model confidence.
H10
تكون عندما يتحسن cross-model continuation memory
بدل conversational. reconstructable
H11
تخفض model text خارج security policy unauthorized mutations
تحت prompt injection.
H12
لا تتبع improvements capability خطًيا؛ improvements reliability
تحتاج optimization منفصلًا.
كل فرضية يجب أن تقبل نتيجة سلبية. إذا لم تظهر فروق بعد عدد كافٍ من seeds/tasks، لا يجوز إنقاذها بتغيير التعريف بعد رؤية النتائج

دون تسجيل ذلك كنسخة جديدة. هذا يحافظ على falsifiability داخل مشروع سريع التطور.
<PARSED TEXT FOR PAGE: 43 / 49>

40. المناقشة: لماذا قد تكون Reliability طريقًا إلى AGI Agentic أكثر من مجرد Safety Layer؟
العمومية العملية لا تتطلب فقط توسيع repertoire المهام، تتطلب الحفاظ على agency عبر بيئات متغيرة وأدوات غير مستقرة ومسارات طويلة. كلما اتسع المجال، زاد عدد حالات الفشل الممكنة. لذلك capability in growth دون reliability قد يرفع سطح الخطأ أسرع من القيمة. من هذه الزاوية، Fabric Reliability ليست قيدًا على AGI بل بنية لتمكين autonomy أعلى بصورة مدروسة.
لكن يجب تجنب استنتاج قوي: نظام يملك rollback/verification ليس AGI. هذه آليات dependability يمكن أن تخدم نظامًا محدودًا أو عامًا. قيمتها في المشروع أنها تجعل الادعاء والاختبار والتنفيذ والذاكرة متصلة، ما يسمح بتقييم أي مكسب capability بطريقة أقل خداعًا.
من ناحية نظرية الحاكمية في MOUSSAIF، الموثوقية تصبح مثالًا واضحًا لحاكمية الحاكمية: planner ليس صاحب الكلمة الأخيرة؛ verifier يحكم على الأثر، authorization يحكم على مشروعية الفعل، recovery يحكم على ما بعد الفشل، وgate memory يحكم على ما يستحق أن يصبح خبرة. النظام لا يراجع النتيجة فقط، يراجع شروط إعطاء السلطة لكل وحدة.
وعليه، يمكن تلخيص التحول من Agent إلى System Agentic Reliable في انتقالين: من »هل أستطيع أن أفعل؟« إلى »هل يحق لي وهل أستطيع إثبات ما فعلت؟«، ومن »تعلمت من الماضي« إلى »تعلمت من ماضٍ ثبتت حالته وحدوده«.
<PARSED TEXT FOR PAGE: 44 / 49>

41. خارطة تنفيذ بحثية لمدة 12 شهرًا
المخرجات المدة المرحلة
P0 Instrumentation 1–2 شهر
+ Trace schema + Evidence Ledger
PostconditionSpec
P1 Mutation Reliability 3–2
+ Reversibility Classifier + Mutation Gate
checkpoints
P2 World Effect 4–3
Independent verifier + state diff + closure
states
P3 Recovery 5–4 Failure Object + rollback/compensation tests
P4 Reliability Bench 7–5
+ R-Suite + repeated runs + perturbations
fault injection
P5 Calibration 8–7 operational confidence + thresholds
P6 Verified Learning 10–8 VEU + skill promotion experiments
P7 Cross-runtime 11–10
MOUSSAIXT adapters + cross-model
continuation
P8 Replication 12–11
frozen baseline + ablation + external
replication package
معيار النجاح في نهاية السنة ليس »عدد components »بل وجود traces dataset قابلة لإعادة التشغيل، benchmarks ثابتة، profile reliability منشور، gains وresults negative، ومجموعة mechanisms التي صمدت أمام ablation. إذا فشلت وحدات، يحذف أو يدمج ما لا يقدم قيمة.

<PARSED TEXT FOR PAGE: 45 / 49>

42. الخلاصة

تدافع الورقة عن انتقال معماري: من Reliability بوصفها نتيجة مرغوبة إلى Reliability بوصفها حاكمية تشغيلية. في هذا التصور، الوكيل لا يملك صلاحية مساواة الخطة بالفعل، ولا الفعل بالأثر، ولا الأثر بالنجاح، ولا النجاح بالتعلم. كل انتقال يحتاج عقدًا ودليلًا يناسب مرتبته.

RFAGI لا يضيف قيمة إذا بقي تسمية. قيمته ستثبت فقط إذا خفض success-false وmutation unsafe، رفع robustness والتعافي، حسّن calibration والذاكرة، وفعل ذلك بتكلفة معقولة وقابلة للتكرار. لذلك صممت الورقة نفسها كبرنامج هدم: state machine، metrics، ablations، benchmarks، للادعاء واضحة وحدود.

المبدأ النهائي المقترح لـMOUSSAIF XT هو: لا تجعل الذكاء يقاس فقط بما يستطيع الوصول إليه، بل بما يستطيع الوصول إليه بصورة متحققة، قابلة للتفسير، محدودة العاقبة، قابلة للإصلاح، وقابلة لأن تتحول إلى معرفة دون تسميم المستقبل.

العقد النهائي

Intent Evidence Authorization Reversible → → → → → → → Plan Action Observation Verification → → Memory Verified Closure/Recovery Effect-World. أي حلقة مفقودة يجب أن تظهر كـrisk residual، لا أن تُخفى داخل كلمة »نجاح«.

<PARSED TEXT FOR PAGE: 46 / 49>

المراجع الداخلية والخارجية

أولًا: مصادر MOUSSAIF Corpus المستخدمة

M1. MOUSSAIF AGI OS / MOUSSAIXT — Deep Engineering Architecture, 50p, 2026. ص صاخصو 3–7، 31–33، 43–40: Claim Discipline، Authorization/Reversibility، World-Effect، Observability، Failure Recovery.

M2. MOUSSAIF Unified AGI Research Book — Arabic RTL/Amiri, 225p, 2026. ص صاخصو 141–143و 188–193: Constitutional Reversibility، Evidence Hierarchy، Trace/Audit، Failure Recovery.

M3. Moussaif QRET — Consciousness / Open Intelligence, 30p, 2026. ص 14: بوابة 22: ص التصعيدية، الموثوقية نظرية الموثوقية وعلاقات الدليل/الهدم.

M4. MOUSSAIFXT vNEXT-3 / v8 engineering activation specification (Pasted markdown, 2026), الأقسام 55–64: Preservation Law، Evolution Lab، Anti-Component-Inflation، Release Scorecard، promotion gates.

M5. Moussaif AGI-Fattah Self-Evolving Capability Federation and Memory OS 05, 2026: constitution for skill orchestration, bounded consequential actions, courts, transfer testing, failure-to-memory policy.

ثانيًا: أدبيات خارجية مقارنة

R1. Rabanser, S.; Kapoor, S.; Kirgis, P.; Liu, K.; Utpala, S.; Narayanan, A. (2026). Towards a Science of AI Agent Reliability. arXiv:2602.16666.

R2. Yao, S.; Shinn, N.; Razavi, P.; Narasimhan, K. (2024). -bench: A τ Benchmark for Tool-Agent-User Interaction in Real-World Domains. arXiv:2406.12045.

R3. Lu, J. et al. (2024). ToolSandbox: A Stateful, Conversational, Interactive Evaluation Benchmark for LLM Tool Use Capabilities. arXiv:2408.04682.

R4. Xie, T. et al. (2024). OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. arXiv:2404.07972.

R5. Debenedetti, E. et al. (2024). AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. arXiv:2406.13352.

R6. Zhang, H. et al. (2024). Agent Security Bench (ASB): Formalizing and Benchmarking Attacks and Defenses in LLM-based Agents. arXiv:2410.02644.

R7. Cuadron, A.; Yu, P.; Liu, Y.; Gupta, A. (2025). SABER: Small Actions, Big Errors — Safeguarding Mutating Steps in LLM Agents. arXiv:2512.07850.

R8. Li, X.; Chen, K.; Long, Y.; Bai, X.; Xu, Y.; Zhang, M. (2025). Generator-Assistant Stepwise Rollback Framework for Large Language Model Agent. arXiv:2503.02519.

.R9. Zhai, Z.; Li, M.; Wang, X. (2026). Revisable by Design: A Theory of Streaming LLM Agent Execution arXiv:2604.23283.
R10. Doshi, A. et al. (2026). Towards Verifiably Safe Tool Use for LLM Agents. arXiv:2601.08012.
.R11. Ruan, Y. et al. (2023). Identifying the Risks of LM Agents with an LM-Emulated Sandbox (ToolEmu) arXiv:2309.15817.
R12. Mialon, G.; Fourrier, C.; Swift, C.; Wolf, T.; LeCun, Y.; Scialom, T. (2023). GAIA: a benchmark for General AI Assistants. arXiv:2311.12983.
/ R13. Yao, S. et al. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. ICLR 2023 arXiv:2210.03629.

ملاحظة على النسب العلمي

المراجع الخارجية تثبت وجود مشكلات ومناهج مرتبطة بالموثوقية، لكنها لا تثبت RFAGI أو 2RAGI. الأسماء والدمج المعماري والصيغ التجريبية في هذه الورقة هي synthesis MOUSSAIF Proposed وتبقى قابلة للهدم. كما أن وجود عنصر في corpus لا يعني أنه deployed أو benchmarked؛ مراتب 0E–5E تبقى الحاكم.

ملحق A — Machine-Readable Contract مختصر

يمكن تحويل المفاهيم إلى schema أولي كالآتي؛ الغرض توجيهي وليس API نهائيًا:

MissionSpec

mission_id; goal; constraints; authorization_tier; risk_features; preconditions; postconditions;
negative_constraints; tool_scope; budget; rollback_requirement; evidence_required;
stabilization_window.

FailureObject

failure_id; category; node; observed_error; evidence; retry_count; blast_radius; candidate_causes;
repair_options; rollback_ref; residual_risk; verifier_status.

VerificationVerdict

verdict_id; action_receipt; source_independence; observed_poststate; intended_poststate; state_diff;
side_effects; closure; confidence; unresolved_gaps.

Verified Experience Unit

veu_id; mission; before_state; trace_refs; evidence_refs; verifier_verdict; failure_or_success;
causal_notes; transfer_scope; tests; promotion_status.

ملحق B — Release Scorecard مقترح لـRFAGI

البعد	RC قبل Minimum evidence
Architecture Integrity	contracts machine-readable + versioned
Skill Reliability	transfer test + known failures
World-Effect Reliability	closure benchmark pass
Rollback	failure injection + verified restore
Observability	trace completeness + replay

Predictability calibration profile
Security المحددة suite في 0 = critical violations
Memory contamination/regression within threshold
Reproducibility multi-run + frozen baseline + seeds
Ablation critical mechanisms show marginal gain
قرار release يجب أن يظل متعدد المحاور. لا تخِف محوًرا ضعيفًا داخل score AGI average. هذا يقتبس مباشرة روح 3vNEXT-
Reliability Fabric. [M4]لـ ويخصصها Release Scorecard